UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/664,531	09/18/2000	Richard Wilson	36.P283	4683

5514 7590 04/16/2004

FITZPATRICK CELLA HARPER & SCINTO
30 ROCKEFELLER PLAZA
NEW YORK, NY 10112

EXAMINER	
ZHEN, LI B	

ART UNIT	PAPER NUMBER
2126	6

DATE MAILED: 04/16/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/664,531	Applicant(s) WILSON ET AL.	
	Examiner Li B. Zhen	Art Unit 2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-57* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-57* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *12 December 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *5*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 – 57 are pending in the application.

Specification

2. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 55 – 57 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

5. Claims 55 – 57 are indefinite because it is unclear whether these are storage medium, computer system, or method claims because they depend on a method claim.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. **Claims 1 – 3, 5 – 7, 9 – 18, 36 – 44 and 55 – 57 are rejected under 35 U.S.C. 102(b) as being anticipated by "The Phoenix Framework: A Practical Architecture for Programmable Networks" (hereinafter Yadav).**

8. As to claim 1, Yadav teaches a method for providing a virtual device container [proactive device object (PDO); left col., p. 3] to virtually extend the functionality of a network device [proactive services are Java objects with well-defined interfaces that provide new functionality; left col., 4th paragraph, p. 2] on a network for supporting a plurality of functional application modules [an agent (an encapsulation of code and state; right col., 1st paragraph, p. 2] residing in a server [proactive console can also be used to created new mobile agents; right col., 2nd paragraph, p. 2] on the network, said method comprising the steps of:

receiving a function request sent from one of the functional application modules [agent first makes a request to the PEnv for the interface of the PDO for the device it wants to operate upon; right col., 1st section, p. 4], the function request corresponding to the network device [mobile agent arrives at the active device; right col., 1st section, p. 4];

selecting one of a plurality of functional component modules in response to the function request [the agent then asks the PDO for the proactive service it needs access to; right col., 1st section, p. 4], each of the functional component modules corresponding to a respective one of the functional application modules [proactive services permit third parties to add a new functionality to an active device; Proactive Service, right col., p. 2],

the selected functional component module corresponding to the functional application module which sent the function request [PDO creates and configures an instance of the requested proactive service and returns a reference to the agent; right col., 1st section, p. 4]; and

executing the selected functional component module according to the function request [agent invokes the proactive service to perform the necessary actions; right col., 1st section, p. 4] wherein each functional component module communicates with the corresponding functional application module through a first interface [Proactive services are objects that expose one or more interfaces; right col., Proactive Services, p. 2] and communicates with the network device through a second interface [PEnv is a Java object that exposes a set of well-defined Java interfaces. These interfaces are the only method available to agents for programming and managing an active device; right col., Proactive Services, p. 2].

9. As to claim 2, Yadav teaches each functional application module implements a different network-wide application [these services can be used to enable network resource management, fault diagnosis, transcoding, and new protocol support; A Programmable Network Framework, left col., p. 2].

10. As to claim 3, Yadav teaches each functional component module provides functional support on behalf of the network device for each corresponding functional application module, respectively [proactive service on the left executes inside the PEnv,

whereas the proactive service on the right interacts with some native library to manipulate the actual device; left col., first paragraph, p. 4].

11. As to claim 5, Yadav teaches one of the separate functional application modules implements a network management application [Scriptable Remote Network Management; p. 5].

12. As to claim 6, Yadav teaches one of the separate functional application modules implements a network security application [Intrusion Detection; p. 6].

13. As to claim 7, Yadav teaches one of the separate functional application modules implements a resource management application [these services can be used to enable network resource management, fault diagnosis, transcoding, and new protocol support; A Programmable Network Framework, left col., p. 2].

14. As to claim 9, Yadav teaches providing a description [repository of information...to manage the device] of each of the plurality of functional component modules for access by each of the plurality of functional application modules [PDO is the repository of information that is needed by PEnv to manage the device. A PDO also contains the configuration data for proactive services; left col., Configuration, p. 3].

15. As to claim 10, Yadav teaches the function request contains a reference to one of the plurality of functional component modules [PDO creates and configures an instance of the requested proactive service and returns a reference to the agent; right col., p. 4].

16. As to claim 11, Yadav teaches the function request is a function call which is supported by one of the plurality of functional component modules [PDO creates and configures an instance of the requested proactive service and returns a reference to the agent; right col., 1st section, p. 4].

17. As to claim 12, Yadav teaches the function request is supported by an operating system component [native library] interoperability standard [the proactive service on the right interacts with some native library to manipulate the actual device; left col., first paragraph, p. 4].

18. As to claim 13, Yadav teaches an operating system registry contains a registry entry corresponding to each of the plurality of functional component modules [proactive console is the central repository for all Java class files representing proactive services and mobile agents; Proactive Console, p. 2].

19. As to claim 14, Yadav teaches the step of loading, by a functional component keeper module [Install and Storage Services; Proactive Services, p. 2], the functional

component module corresponding to each registry entry for the network device in the operating system registry [Install and Storage Services are invoked by the proactive console to persistently install new services on an active device; Proactive Services, p. 2].

20. As to claim 15, Yadav teaches the loading step is performed in response to an initialization command [Install and Storage Services are invoked by the proactive console to persistently install new services on an active device; Proactive Services, p. 2].

21. As to claim 16, Yadav teaches the function request is a request for information [fault diagnosis] from the network device [these services can be used to enable network resource management, fault diagnosis, transcoding, and new protocol support; A Programmable Network Framework, left col., p. 2].

22. As to claim 17, Yadav teaches the function request is a request for the network device to receive information [these services can be used to enable network resource management, fault diagnosis, transcoding, and new protocol support; A Programmable Network Framework, left col., p. 2].

23. As to claim 18, Yadav teaches the function request is a request for the network device to perform a function [agent invokes the proactive service to perform the necessary actions; right col., 1st section, p. 4].

24. As to claims 36 and 37, Yadav teaches the second interface is the network and the second interface is a serial bus [Interfaces provides by PEnv can be invoked locally or remotely using Java Remote Method Invocation (RMI); right col., Proactive Environment, p. 2].

25. As claims 38 and 39, Yadav teaches each functional component module reads data from a memory of the network device via the second interface [proactive service can interact with a device locally via a native library (provides access to device resources that cannot be accessed directly for Java); left col., p. 3].

26. As to claim 40, Yadav teaches one of the functional application modules is a proxy application [proxy for other devices] which provides a data interface over the network between the plurality of functional component modules and a third-party application [PEnv may contain more than one PDO when the active device is acting as a proxy for other devices (legacy, non-active). Whenever a network device needs to be managed, the proactive console sends the PDO for the device either to the device itself...or to some other active device that acts as a proxy for managing the actual device; Configuration, p. 3].

27. As to claim 41, Yadav teaches the function request is a generic request which is supported in the selected functional component module by a plurality of specific protocol requests, and wherein one of the plurality of specific protocol requests is sent from the selected functional component module to the network device based on a desired protocol for communication with the network device [proactive service can interact with a device locally via a native library (provide access to device resources that cannot be accessed directly fro Java) or remotely via Remote Procedure Call (RPC) mechanisms such as Java Remote Method Invocation (RMI), Microsoft Distributed Component Model (DCOM), and Common Object Broker Architecture (CORBA); left col., p. 3].

28. As to claim 42, Yadav teaches a second plurality of functional component modules [proactive services are Java objects with well-defined interfaces that provide new functionality; left col., 4th paragraph, p. 2] are used to support a second network device [networks are composed of heterogeneous devices; right col., p.4], and wherein each functional application module is supported by the corresponding functional component module for each network device [PDO creates and configures an instance of the requested proactive service and returns a reference to the agent; right col., 1st section, p. 4].

29. As to claim 43, Yadav teaches the virtual device container is a DCOM server [Microsoft Distributed Component Model (DCOM); left col., p. 3].

30. As to claim 44, Yadav teaches the function request is addressed to the virtual device container [the agent then asks the PDO for the proactive service it needs access to; right col., 1st section, p. 4].

31. As to claims 55 – 57 [note the 35 USC 112, second paragraph rejection above], Yadav teaches a network computing device for providing a virtual device container to virtually extend the functionality of a network device on a network for supporting a plurality of functional application modules residing in a server on the network [PEnv may contain more than one PDO when the active device is acting as a proxy for other devices (legacy, non-active). Whenever a network device needs to be managed, the proactive console sends the PDO for the device either to the device itself...or to some other active device that acts as a proxy for managing the actual device; Configuration, p. 3], comprising a program memory for storing process steps executable to perform a method according to claim 1 and a processor for executing the process steps stored in said program memory [PEnv is running low on some critical resource such as CPU or memory; left col., p. 4].

Claim Rejections - 35 USC § 103

32. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the

invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

33. **Claims 4 and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yadav in view of U.S. Patent NO. 5,926,539 to Shtivelman.**

34. As to claims 4 and 8, Yadav does not teach function application modules that implement an e-mail application.

35. However, Shtivelman teaches an active network and extending the network to include new capabilities such as e-mail [capability is extended to multimedia communication, such as e-mails, video mails and the like; col. 4, lines 7 – 20].

36. It would have been obvious to a person of ordinarily skilled in the art at the time of the invention to apply the teaching of applications modules that implement e-mail functionalities as taught by Shtivelman to the invention of Yadav because this provides an efficient means of distributing information internal to and external from an enterprise or business.

37. **Claims 19 – 35 and 45 – 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yadav in view of U.S. Patent NO. 6,073,184 to Couturier.**

38. As to claim 19, Yadav teaches providing interfaces between application modules Proactive Services, p. 2] but does not specifically identify the interface as a software bus.

39. However, Couturier teaches a method of transmitting notification in a distributed-application data processing network [col. 2, lines 11 – 29] and a software bus that enables objects to send and receive requests in a distributed environment [col. 5, lines 28 - 50; col. 6, lines 20 - 30].

40. It would have been obvious to a person of ordinarily skilled in the art at the time of the invention to apply the teaching of a software bus as taught by Couturier to the invention of Yadav because this deliver requests to objects concerned and to return output values to client objects in transparent manner without the client object knowing where the objects are located in the network, how they are implemented, how they are stored, nor how they are executed [col. 1, lines 39 - 49 of Couturier].

41. As to claim 20, Yadav as modified teaches the dedicated software bus is managed by a software bus control module [notification service is a set of CORBA objects which operate on the software bus; col. 6, lines 22 - 30 of Couturier].

42. As to claim 21, Yadav as modified teaches each functional component module [proactive services permit third parties to add a new functionality to an active device; Proactive Service, right col., p. 2 of Yadav] and each functional application module contain a software bus connector module which supports communication over the dedicated software bus [Each of the notification channels 32 and 34 is specific to a particular notification category. In the example under consideration, it is assumed that notification channel 32 is specific to notifications concerning black-and-white printing

while notification channel 34 is specific to notifications concerning color printing; col. 6, lines 30 – 39 of Couturier].

43. As to claim 22, Yadav as modified teaches communication over the dedicated software bus between the software bus connector modules [Each of the notification channels 32 and 34 is specific to a particular notification category; col. 6, lines 30 – 39 of Couturier] is implemented by using a plurality of different software bus message [proactive service can interact with a device locally via a native library (provide access to device resources that cannot be accessed directly fro Java) or remotely via Remote Procedure Call (RPC) mechanisms such as Java Remote Method Invocation (RMI), Microsoft Distributed Component Model (DCOM), and Common Object Broker Architecture (CORBA); left col., p. 3 of Yadav].

44. As to claim 23, Yadav as modified teaches one of the software bus messages is an information request from one of the functional application modules for identification information [identifier] corresponding to one of the functional component modules [first notification service addresses a registration message to the second notification service...the second notification service returns an identifier of the second notification service to the first notification service; col. 2, lines 49 – 60 of Couturier].

45. As to claim 24, Yadav as modified teaches in response to the information request, identification information corresponding to the requested functional component

module is sent to the requesting functional application module [first notification service stores said identifier of the second notification service for subsequent transmission of notifications to the second notification service; col. 2, lines 49 – 60 of Couturier].

46. As to claim 25, Yadav as modified teaches the requesting functional application module establishes a direct communication connection with the requested functional component module by using the identification information [After the IDL interface has been compiled, the resulting stub is linked to the implementation of the object; col. 6, lines 10 – 13 of Couturier].

47. As to claim 26, Yadav as modified teaches the direct communication connection is over a data channel of the dedicated software bus [Each of the notification channels 32 and 34 is specific to a particular notification category; col. 6, lines 30 – 39 of Couturier].

48. As to claim 27, Yadav as modified teaches the direct communication connection is implemented in one of a plurality of different communication protocols [proactive service can interact with a device locally via a native library (provide access to device resources that cannot be accessed directly fro Java) or remotely via Remote Procedure Call (RPC) mechanisms such as Java Remote Method Invocation (RMI), Microsoft Distributed Component Model (DCOM), and Common Object Broker Architecture (CORBA); left col., p. 3 of Yadav].

49. As to claim 28, Yadav as modified teaches one of the software bus messages is an information request for identification information [identifier] corresponding to all functional component modules and functional application modules that have software bus connector modules [first notification service addresses a registration message to the second notification service...the second notification service returns an identifier of the second notification service to the first notification service; col. 2, lines 49 – 60 of Couturier].

50. As to claim 29, Yadav as modified teaches one of the software bus messages is an event message for notifying all functional component modules and functional application modules of a software bus event [notification is transmitted from the emitter object to the receiver object via the first and second notification services in succession at the initiative of the emitter object; col. 9, lines 29 – 33 of Couturier].

51. As to claims 30 – 33, Yadav as modified teaches the software bus event is a reset event, a startup event, a shutdown event, and a pause event [emitter objects are associated with printers and are suitable for addressing notifications concerning the operating state of a printer, in particular whether the printer is available or unavailable because of a breakdown. The receiver objects are assumed to be objects handling print functions and associated with workstations; col. 5, line 63 - col. 6, line 5 of Couturier].

52. As to claim 34, Yadav as modified teaches one of the software bus messages is a subscription message from one of the software bus connector modules [first notification service addresses a registration message to the second notification service...the second notification service returns an identifier of the second notification service to the first notification service; col. 2, lines 49 – 60 of Couturier], wherein, in response to the subscription message, the dedicated software bus subsequently passes all software bus messages of a specified type to the requesting software bus connector module [first notification service stores said identifier of the second notification service for subsequent transmission of notifications to the second notification service; col. 2, lines 49 – 60 of Couturier].

53. As to claim 35, Yadav as modified teaches the software bus messages are supported by a plurality of different communication protocols [proactive service can interact with a device locally via a native library (provide access to device resources that cannot be accessed directly fro Java) or remotely via Remote Procedure Call (RPC) mechanisms such as Java Remote Method Invocation (RMI), Microsoft Distributed Component Model (DCOM), and Common Object Broker Architecture (CORBA); left col., p. 3 of Yadav].

54. As to claim 45, Yadav as modified teaches the virtual device container registers an entry with the dedicated software bus [new proactive services are installed and registered with the proactive environment on an active device with the help of the core

proactive services; left col., Installation, p. 3 of Yadav], the entry containing a node address corresponding to the virtual device container [when the PEnv is started, it is given the address of the proactive console so that the PEnv can download the additional services from the proactive console; left col., Installation, p. 3 of Yadav].

55. As to claim 46, Yadav as modified teaches the virtual device container obtains a globally unique identifier through an operating system function call [object references which form the identifiers of the objects under consideration are replaced by identifiers for the software components under consideration, e.g. addresses; col. 5, lines 55 – 65 of Couturier].

56. As to claim 47, Yadav as modified teaches each of the functional application modules has access to the globally unique identifier from the virtual device container by using the node address [when the PEnv is started, it is given the address of the proactive console so that the PEnv can download the additional services from the proactive console; left col., Installation, p. 3 of Yadav], whereupon a direct software connection is available to the virtual device container by using the globally unique identifier [object references which form the identifiers of the objects under consideration are replaced by identifiers for the software components under consideration, e.g. addresses; col. 5, lines 55 – 65 of Couturier].

57. As to claim 48, Yadav as modified teaches at least one of the functional application modules establishes a direct software connection to the virtual device container by using the node address of the virtual device container [object references which form the identifiers of the objects under consideration are replaced by identifiers for the software components under consideration, e.g. addresses; col. 5, lines 55 – 65 of Couturier].

58. As to claims 49 – 51, Yadav as modified teaches the direct software connection is supported by COM function calls, JAVA function calls, and CORBA function calls [proactive service can interact with a device locally via a native library (provide access to device resources that cannot be accessed directly fro Java) or remotely via Remote Procedure Call (RPC) mechanisms such as Java Remote Method Invocation (RMI), Microsoft Distributed Component Model (DCOM), and Common Object Broker Architecture (CORBA); left col., p. 3 of Yadav].

59. As to claim 52, this is rejected for the same reasons as claim 46 above.

60. As to claim 53, Yadav as modified teaches the function request is received over a direct software connection between the corresponding functional application module and the virtual device container [agent first makes a request to the PEnv for the interface of the PDO for the device it wants to operate upon; right col., 1st section, p. 4 of Yadav], the direct software connection being established based on the globally

unique identifier [object references which form the identifiers of the objects under consideration are replaced by identifiers for the software components under consideration, e.g. addresses; col. 5, lines 55 – 65 of Couturier].

61. As to claim 54, Yadav as modified teaches a method for providing a virtual device container [proactive device object (PDO); left col., p. 3 of Yadav] to virtually extend the functionality of a network device [proactive services are Java objects with well-defined interfaces that provide new functionality; left col., 4th paragraph, p. 2 of Yadav] on a network for supporting a plurality of functional application modules [an agent (an encapsulation of code and state; right col., 1st paragraph, p. 2 of Yadav] residing in a server [proactive console can also be used to created new mobile agents; right col., 2nd paragraph, p. 2 of Yadav] on the network, said method comprising the steps of:

loading, by a functional component keeper module [Install and Storage Services; Proactive Services, p. 2 of Yadav] in the virtual device container, a plurality of functional component modules corresponding to a plurality of registry entries in an operating system registry [Install and Storage Services are invoked by the proactive console to persistently install new services on an active device; Proactive Services, p. 2 of Yadav], each of the functional component modules corresponding to a respective one of the functional application modules [PDO creates and configures an instance of the requested proactive service and returns a reference to the agent; right col., 1st section, p. 4 of Yadav];

establishing a direct connection between a requesting one of the functional application modules and the virtual device container over a dedicated software bus [After the IDL interface has been compiled, the resulting stub is linked to the implementation of the object; col. 6, lines 10 – 13 of Couturier] by using a globally unique identifier which corresponds to the virtual device container and which is obtained from the virtual device container via the dedicated software bus [object references which form the identifiers of the objects under consideration are replaced by identifiers for the software components under consideration, e.g. addresses; col. 5, lines 55 – 65 of Couturier];

receiving, over the direct connection, a function request sent from the requesting functional application module [agent first makes a request to the PEnv for the interface of the PDO for the device it wants to operate upon; right col., 1st section, p. 4 of Yadav], the function request corresponding to the network device and containing a function call [mobile agent arrives at the active device; right col., 1st section, p. 4 of Yadav];

selecting one of a plurality of functional component modules for supporting the function call [the agent then asks the PDO for the proactive service it needs access to; right col., 1st section, p. 4 of Yadav], the selected functional component module corresponding to the requesting functional application module [PDO creates and configures an instance of the requested proactive service and returns a reference to the agent; right col., 1st section, p. 4 of Yadav]; and

executing the selected functional component module according to the function call [agent invokes the proactive service to perform the necessary actions; right col., 1st

section, p. 4 of Yadav], wherein each functional component module communicates with the network device through the network [Interfaces provides by PEnv can be invoked locally or remotely using Java Remote Method Invocation (RMI); right col., Proactive Environment, p. 2 of Yadav].

Conclusion

62. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent NO. 6,694,363 to Yamadaji teaches a DCM/FCM manager that detects the coincidence of modules by comparing a device control module and a device function module transferred from a DCM/FCM memory of other device with a content in the inside of a DCM/FCM memory of its own device.

U.S. Patent NO. 6,389,466 to Zondag teaches management functionality in a consumer electronic system.

U.S. Patent NO. 6,466,984 to Naveh teaches method for policy-based management of quality of service treatments of network data traffic flows by integrating policies with application programs.

63. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (703) 305-3406. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Li B. Zhen
Examiner
Art Unit 2126

lbz
April 8, 2004

MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100